CirdoX: an On/Off-line Multisource Speech and Sound Analysis Software

F Aman, Michel Vacher, François Portet, W Duclot, Benjamin Lecouteux

CirdoX: an On/Off-line Multisource Speech and Sound Analysis Software

F. Aman, M. Vacher, F. Portet, W. Duclot, B. Lecouteux
CNRS, LIG, F-38000 Grenoble, France
Univ. Grenoble Alpes, LIG, F-38000, Grenoble, France
Frédéric.Aman@imag.fr, Michel.Vacher@imag.fr, Francois.Portet@imag.fr

Abstract

Vocal User Interfaces in domestic environments recently gained interest in the speech processing community. This interest is due to the opportunity of using it in the framework of Ambient Assisted Living both for home automation (vocal command) and for call for help in case of distress situations, i.e. after a fall. CIRDOX, which is a modular software, is able to analyse online the audio environment in a home, to extract the uttered sentences and then to process them thanks to an ASR module. Moreover, this system perfoms non-speech audio event classification; in this case, specific models must be trained. The software is designed to be modular and to process on-line the audio multichannel stream. Some exemples of studies in which CIRDOX was involved are described. They were operated in real environment, namely a Living lab environment.

Keywords: audio and speech processing, natural language and multimodal interactions, Ambient Assisted Living (AAL).

1. Introduction

Voice User Interfaces (VUI) are becoming more and more popular in ambient intelligence environments whether it is for smartphones or for smart homes (e.g., S.A.H.R.A.[1]). For instance, in domestic environments, VUI recently gained interest in the speech processing community as exemplified by the rising number of smart home projects that consider Automatic Speech Recognition (ASR) in their design (Istrate et al., 2008; Badii and Boudy, 2009; Hamill et al., 2009; Filho and Moir, 2010; Gemmeke et al., 2013; Christensen et al., 2013; Cristoforetti et al., 2014; Vacher et al., 2015b).

However, the speech technology, though mature is still difficult to master for non-expert and to tune for a particular usage (Vacher et al., 2015a). Although many good quality commercial systems were made available during this decade, these are neither open source nor customisable by the user. Moreover, for some of them, respect to user privacy may be questioned. Hence, it becomes important for research in natural language interaction to access open-source and highly configurable real-time VUI software.

Many toolboxes and libraries exist for audio processing such as *Planet CCRMA at Home* (CCRMA, 2016), *ChucK* (ChucK, 2016), etc. But these are dedicated for music processing and not for speech interaction. General purpose recorder and editor such as *SoX* (Sox, 2016) and *Audacity* (Audacity, 2016) make it possible to acquire audio signals in real-time and edit them, but these tools are mainly designed to perform off-line processing. Regarding speech interaction, there are a high number of ASR (Automatic Speech Recognition) systems[2] but again, using them in an on-line hand free setting requires a high expertise in signal and speech processing.

Ideally, a complete hand free VUI system would be composed of five main components:

1. an multisource acquisition stage;
2. a signal enhancement stage;
3. a Voice Activity Detection (VAD) stage;
4. an Automatic Speech Recognition (ASR) stage;
5. a Natural Language Understanding (NLU) stage, when required;
6. a decision stage;
7. and, a communication stage.

A few tools have been built to perform some of these stages such as AuditHIS (Vacher et al., 2010) or more recently PATSH (Vacher et al., 2013). They were built to be used in continuous sensing uncontrolled ambient system using a pipeline architecture of audio acquisition, sound detection, classification and ASR. However, these systems are not open-source and difficult to access. In this paper, we present an open-source VUI software, called CIRDOX which performs multisource audio analysis to extract speech occurrences and handle the speech recognition. Moreover, the system performs non-speech audio event recognition (cough, shock, etc.) so that interaction can also be performed using sound. The software is designed to be modular and to process on-line the audio stream (as soon as audio sample chunks arrive). In this way, inclusion of new modules is designed to be easy.

2. CirdoX: an Open Source Audio Analysis Software

The architecture of the presented system is depicted Figure 1. It has been entirely developed in C and it is currently running on Unix-like POSIX systems only. We plan to diffuse the software under the GNU GPL license. The source code will be available on `https://forge.imag.fr/projects/cirdox/`.

The backbone of CIRDOX is its decomposition into three processes running in parallel:

1. The **Acquisition and Detection** Process which handles the continuous audio sensing and the extraction of audio events of sufficient energy from multichannel microphone or record (1 to 8 channels);

[1] `http://sarah.encausse.net`

[2] `https://en.wikipedia.org/wiki/List_of_speech_recognition_software`

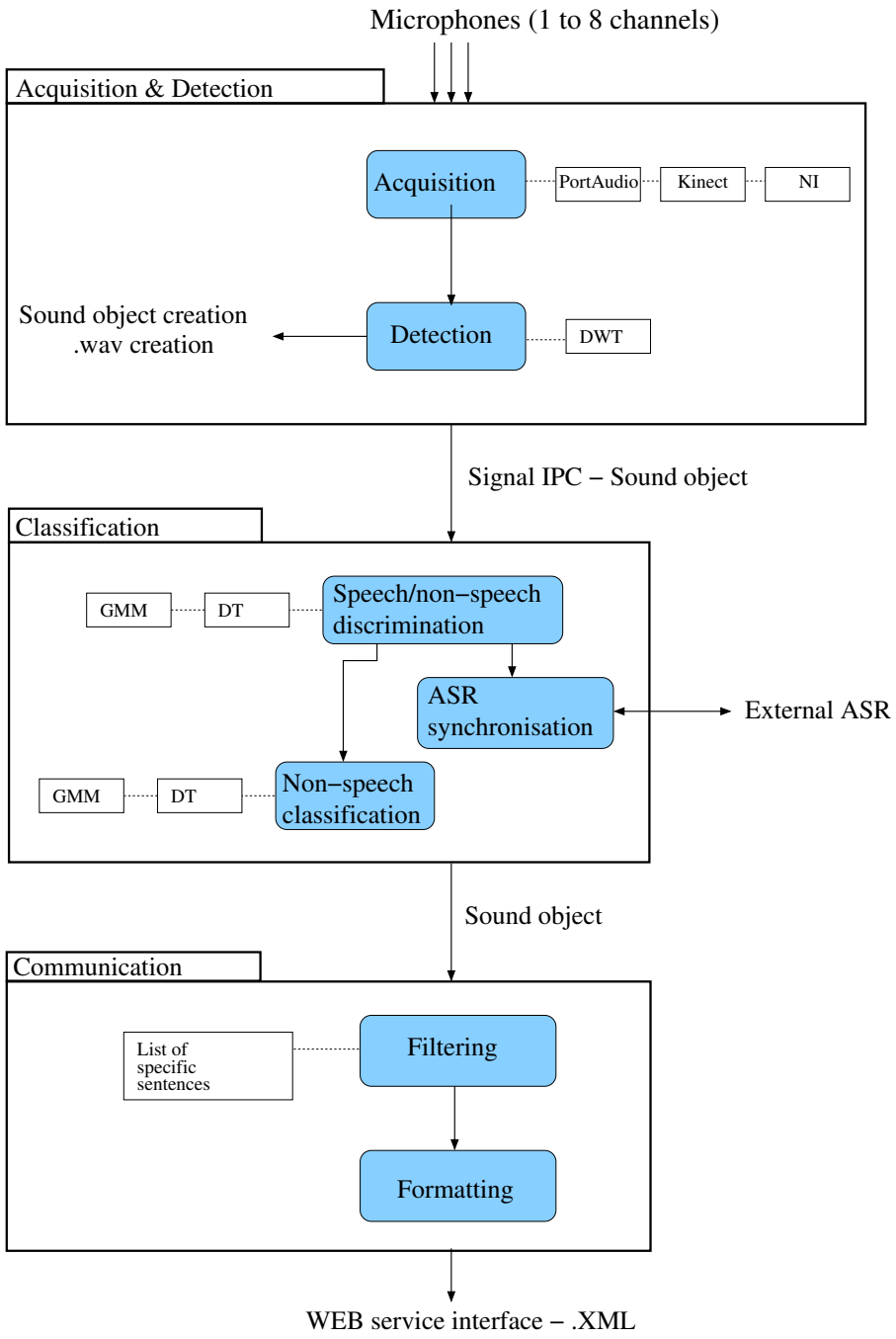

Figure 1: *Architecture of the CirdoX software.*

2. The **Classification** Process, which performs first the speech/non-speech discrimination and secondly the identification of each non-speech event. Thirdly, for speech events, a call to an external ASR system is made by sending speech signal and receiving hypothesis;

3. and, the **Communication** Process, which performs subsequent processing and formatting to communicate, the interpreted audio events to external devices (home automation system, logger, alarm system, etc.).

The CIRDOX application is designed to be modular and configurable, with independent modules so that each module can be chosen from among several plugins corresponding to different ways of achieving a task (for instance, classification using either Gaussian Mixture Model or decision trees). The plugins are loaded using dynamic library linking so that any user can build her own plugins to perform some of the module tasks. Table 1 shows the different plugins implemented and that we plan to implement.

The information exchange is mainly performed using a sound object data structure which contains the full information necessary to characterize a detected audio event (time of begin/end, signal to noise ratio) and whose fields are filled in along the processing chain (type of sound: speech or non-speech, ASR hypothesis, every day sound class, etc.). These sound objects are exchanged between processes using IPC (Inter-Process Communication) sig-

Process	Functionality & modules	Implemented	Library & model
Acquisition & Detection			
	Acquisition: NI on-line	Not yet	NI-DAQmx driver
	Acquisition: NI off-line	Not yet	NI-DAQmx driver
	Acquisition: Portaudio on-line	Yes	libportaudio2 v19 portaudio19dev v19
	Acquisition: Portaudio off-line	Yes	libportaudio2 v19 portaudio19dev v19
	Acquisition: Kinect on-line	Yes	libfreenect v0.2
	Acquisition: Kinect off-line	Not yet	–
	Detection: DWT	Yes	(Vacher et al., 2004)
Classification			
	Speech/non-speech discrimination: GMM	Yes	LIA_RAL v3.0- to be adapted to home audio environment
	Speech/non-speech discrimination: GMM	Yes	ALIZE API v3.0 - to be adapted to home audio environment
	Non-speech classification: GMM	Yes	ALIZE API v3.0 - to be adapted to home audio environment
	Non-speech classification: DT	Not yet	–
	ASR synchronisation: Sphinx3	Yes	to be adapted to the application
	ASR synchronisation: Google API	Yes	different languages supported
	ASR synchronisation: KALDI python server	Yes	to be adapted to the application
Communication			
	Filtering: Levenshtein Distance	Yes	list of sentences to furnish
	Formatting	Yes	to be adapted to the application

Table 1: Functionalities and Implementations of CIRDOX modules.

nals and a common local repository which contain the raw audio data (written by the **Acquisition and Detection** process and read by the **Classification**).
CIRDOX can be executed in two modes:

1. online, meaning the input sources are the different microphones,
2. off-line, meaning that the input is actually extracted from audio files.

The second mode is useful for the development of new modules in order to test them in reproducible conditions. Moreover, the off-line mode can be executed either in real-time (chunks of data are processed at the sampling frequency of the audio record) or in stream mode (chunks of data are processed as fast as possible).

2.1. Acquisition/Detection Process

In online mode, the audio stream is captured by a set of microphones and recorded by the Acquisition module. At this time of writing, this module can be executed through the following plugins:

1. the PortAudio library[3];
2. The Kinect API version 0[4] ;
3. the driver of a National Instrument card[5].

The acquisition can also store the full recorded signal for further analysis as well as simulate an acquisition from a audio file for development and debugging purpose. Once the acquisition has reached a certain amount of data, a call-back function triggers the Detection module.
The Detection module detects the occurrences of audio events (speech or non-speech). It keeps the state of the previous detection between the successive calls by the acquisition module. Rather than multi-threading the detection for each channel, the detection is performed successively on each channel. This is made possible by the synchronous multichannel acquisition. At the time of writing only one plugin as been written to perform the module task. This plugin is based on the change of energy level of the three highest frequency coefficients of the Discrete Wavelet Transform (DWT) in a floating window frame (last 2048 samples without overlapping) (Vacher et al., 2004). Each time the energy on a channel goes beyond a self-adaptive threshold, an audio event is detected until the energy decreases below this level for at least an imposed duration.

[3] `www.portaudio.com` is used by many open-source applications including Audacity to access the sound card

[4] `http://openkinect.org/`

[5] `http://www.ni.com/product-documentation/11787/en/`

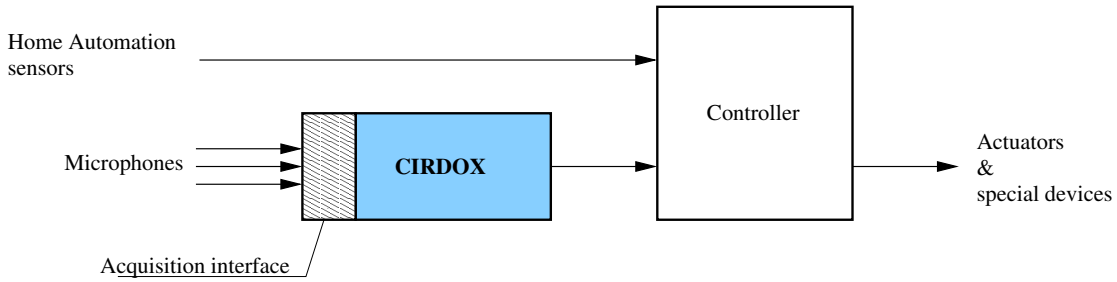

Figure 2: Home Automation system using CirdoX.

Hence, this plugin adapts itself to changes in background noise level.
The module also identifies simultaneous events using simple temporal overlapping[6]. This make it possible to perform multisource processing or data fusion. Too lengthy audio events can also be discarded from the process[7]. During detection, the Signal to Noise Ratio (SNR) of the event is calculated for each channel. Once a detection is complete, event data are saved in an audio file in a local repository. In the case of simultaneous events, the saved data are those from the event with the best SNR. A sound object with a link to the audio file is created and sent by IPC to the next process.

2.2. Classification Process

This process is in charge of interpreting the audio events. A first module, called the Discrimination module, determines whether the audio event is a speech or non-speech event. This module can be executed through plugins using Gaussian Mixture Models (GMM) or Decision Tree (DT) (the DT plugin is not yet implemented). Two plugins with GMM are implemented and use the ALIZE library (Bonastre et al., 2005): the first uses features in the SPRO3 format and integrates ALIZE API functions, and the second uses features in the SPRO4 format and calls LIA_RAL tools. In both cases, signal discrimination is achieved through a Gaussian Mixture Model (GMM) classifier, trained with a everyday life sound corpus, and a speech corpus recorded in the LIG laboratory (Istrate et al., 2006). Acoustical features can be Linear-Frequency Cepstral Coefficients (LFCC) or Mel-Frequency Cepstral Coefficients (MFCC). The classifier uses 24 Gaussian models, on a logarithmic Mel scale.
In the case of non-speech events, the Classification module attributes a sound class (e.g., cough, shock, etc.) to the event. Again, some plugins based on GMMs, Decision Tree (DT) are or will be available. For the moment, only the GMM plugin, with a classifier using the ALIZE API functions, is implemented. In the case of speech, the signal can be sent to an external ASR (Automatic Speech Recognition) system so that a transcription of the speech can be obtained. For the ASR synchronisation task, several plugins can be used, corresponding to different ASR systems: Sphinx3, Google Speech API and KALDI.

[6] An event might be captured by several microphones, for instance when they are set in the same room.

[7] Depending on the application, sound events might have a minimum and maximum duration. This make it possible to filter out too short events and installed background noise.

Regarding the plugin connecting to Sphinx3, ASR is managed with a Tcl/Tk script that calls Sphinx3 tools[8] for signal to MFCC conversion and decoding. The synchronization between the Classification process and the Tcl/Tk script is done by lock files. A text file with the hypothesis is generated by Sphinx3.
Regarding the plugin connecting to Google Speech API, the wav file is converted into the FLAC format and sent to the Google Speech API server[9] through a POST request. The Google server return a JSON response with the hypothesis.
Thanks to the plugin connecting to KALDI, the wav file is sent to an ASR server, based on the Kaldi toolkit (Povey et al., 2011) and the GStreamer framework, and implemented in Python[10]. Like the Google Speech API plugin, a JSON response with the hypothesis is returned.
Sound classification results or ASR hypothesis are included in the sound object for processing by the next module.

2.3. Communication Process

The communication process handles the interaction with the exterior. It also logs all the sound events. This part is highly dependant on the application to communicate with. To take this dependence into account a filter module can be activated to communicate only the event to communicate to the exterior (e.g., only speech of a certain kind) and/or to perform more processing on the event. The formatting module takes care only of the communication protocol.

3. CirdoX in Ambient intelligence Setting

CIRDOX has been used at different occasions to capture and analyse audio events in different smart spaces including the smart class room of the MSTIC Fab-Lab and the Domus smart home of the LIG. We present below the results of a setting for voice command recognition.

3.1. Voice command recognition for Home Automation

Figure 2 shows the connections of CIRDOX with the input (wireless microphones set in rooms of a home) and the output (home automation system) that has be employed both in the CIRDO project (Bouakaz et al., 2014) and the sweet-Home project (Vacher et al., 2015b). The aim is to constantly capture the audio signal to seek for specific voice

[8] http ://cmusphinx.sourceforge.net

[9] http://www.google.com/speech-api/v1/recognize

[10] https://github.com/alumae/kaldi-gstreamer-server

commands and send them to the controller of the home automation system in order to perform the commands when applicable. In the case of a generic smart home equipped with a home automation system, actuators are used to activate lighting, stores, multimedia devices, etc. In the particular case of calls for help, the only action is alarm sending to a specialized service or to a relative. In all cases, the speaker is distant from the microphone (i.e., no worn microphone) and the interaction is hand-free (no push button to start the recording).
To address voice command recognition realistically, 3 kinds of speech were considered:

- vocal command (e.g., "home turn on the light"),
- distress calls (e.g., "home call for help")
- or colloquial speech (e.g., "I ate spinach at lunch").

In this last case, no action must be operated and the audio event must be discarded. Therefore, it is necessary to check if a speech audio event corresponds to a voice command or a distress call. To do so, the exhaustive list of all possible voice commands and distress calls was generated wrt a predefined grammar. Since a word-by-word comparison would be too crude, each speech transcription hypothesis was first phonetized and then a distance was computed against the list of phonetized voice commands. Formally, for each phonetized ASR output T, every voice commands H is aligned to T using Levenshtein distance (Levenshtein, 1966). The deletion, insertion and substitution costs were computed empirically while the cumulative distance $\gamma(i, j)$ between H_j and T_i is given by Equation 1.

$$\gamma(i,j) = d(T_i, H_j) + min\{\gamma(i-1, j-1), \gamma(i-1, j), \gamma(i, j-1)\} \quad (1)$$

The sentence with the aligned symbols score is then selected for decision according a detection threshold. This approach takes into account some recognition errors such as word endings or light variations. Moreover, in a lot of cases, a miss-decoded word is phonetically close to the true one (due to the close pronunciation). At the end of the process, the sound object is received by the Data Formatting module that sends a socket containing all the relevant data to the controller system.

3.2. Experiment in Living lab

CIRDOX has been used in the framework of the CIRDO project with participants who played scenarios in the Living lab DOMUS, including falls down on the floor and calls for help uttered in French. Since it is difficult to find elderly people capable and accepting to play such scenarios, some recruited people were under 60. Overall 17 French participants were recruited (9 men and 8 women) and 4 were elderly people (Vacher et al., 2016). The younger participants were instructed to wear an equipment i.e. old age simulator. Old age simulator hampered mobility, reduced vision and hearing. Figure 3 shows a young participant during the training phase before playing a fall scenario on the carpet. Scenarios included call for help sentences. Table 2 gives some examples of these sentences. The size of audio records is 645.54 seconds.

Distress Sentence	AD80 Sentence
Aïe aïe aïe	Aidez-moi
Oh là	Au secours
Merde	e-lio, appelle du secours
Je suis tombé	e-lio appelle ma fille
Je peux pas me relever	Appelle quelqu'un e-lio
Qu'est-ce qu'il m'arrive	e-lio, appelle quelqu'un
Aïe ! J'ai mal	e-lio appelle ma fille
Oh là ! Je saigne ! Je me suis blessé	e-lio appelle le SAMU

Table 2: Some examples of French sentences identified for inclusion in the scenarios

Figure 3: A participant during the training phase of a fall scenario (Bouakaz et al., 2014).

3.3. CIRDOX Configuration

CIRDOX was used in order to process data and to extract online the sentences uttered during the experiment.
GMM models were trained using the ALIZE software in order to adapt the Speech/Non-speech Discrimination module. Training data were made of distant speech records from SWEET-HOME corpus (Vacher et al., 2014) recorded in the Living lab DOMUS: 9,147 speech files (home automation orders or distress calls) were used to train the speech model (i.e. 4 hours 53 minutes), 13,448 sound files (i.e. 5 hours 10 minutes) were used to train the non-speech model.
ASR was operated using Sphinx3 with an acoustic model trained on the BREF120 corpus (Lamel et al., 1991) and adapted to elderly voice, to distant speech and to expressive speech conditions. BREF120 is a large corpus of oral read texts, containing more than 100 hours of speech produced in French by 120 speakers (65 women and 55 men), all recorded texts were extracted from the newspaper *Le Monde*. The adaptation was obtained thanks to the MLLR method with the "User Specific Interaction" subset of the SWEET-HOME corpus (Vacher et al., 2014) which is made of 337 sentences in French (9min 30s of signal) uttered in the DOMUS flat by elderly or visually impaired people (average age: 72 years), and with 742 expressive speech sen-

Type of sound	Classified as speech	Classified as non-speech
Speech	237	85
Non-speech	7	1621

Table 3: Discrimination between "Speech" and "Non-speech".

tences of the "Voix détresse" corpus recorded by the LIG laboratory.

A language model specific to sentences to recognize was estimated from distress sentences and calls to caregivers of AD80 corpus (Vacher et al., 2015a). This is a trigram model containing 88 unigrams, 193 bigrams and 223 trigrams. A general language model has been learned from the French Gigaword corpus (Linguistic Data Consortium, 2011) which is a collection of newspaper articles that have been acquired over several years by the Linguistic Data Consortium (LDC) from the University of Pennsylvania. This model is unigram and contains 11018 words. The final model is the combination of the specific model with the general model, giving more weight (90%) on the probabilities of specific model.

3.4. Audio Analysis Results

The Detection module extracted 1,950 sound events. The confusion matrix of the results of the Speech/Non-speech Discrimination module are given in Table 3. This module classified 244 of the events as speech, 237 were really speech, 7 were non-speech events classified as speech. 85 speech events were badly classified as sound. At the output of the Speech/Non-speech Discrimination module, the specificity is high, $Sp = 0.996$, this parameter represents the classification rate of non-speech events classified as non-speech. However, the sensitivity is low, $Se = 0.736$, this parameter represents the classification rate of speech events as speech.

According to results of previous stages of CIRDOX, 244 audio events were decoded, 204 are distress calls, 33 colloquial sentences and 7 non-speech signal. Table 4 presents the decoding results. WER is very high for colloquial sentences, this is due to the language model which is non adapted to this task and corresponds to a desired effect: this kind of sentences is not to be well recognized for privacy reason. Regarding distress calls, the Word Error Rate (WER) is quite high with 38.52%. This result is far from perfect but they were obtained under harsh conditions. Indeed, the participants played scenarios which included falls on the floor: they generated a lot of noise sounds which were often mixed with speech, and they were far from the microphone.

At the output of the filtering stage, 29 colloquial sentences and sounds were rejected (True Negative), 35 distress call were rejected by mistake (False Negative), 11 colloquial sentences and sounds were kept by mistake (False Positive), and 169 distress calls were recognized (True Positive). Among the 204 distress calls, 82.84% of the calls are well detected. Table 5 summarize the performances of this stage. The False Alarm Rate (FAR) is 6.11%.

Type	WER (%)
Calls for help	38.52
Other (colloquial and non-speech)	115.31

Table 4: WER for the DOMUS recorded data.

Performance measure	Value (%)
Se	82.84%
Sp	72.50%
FAR	6.11%

Table 5: Sensitivity, specificity and false alarm rate of the filtering stage from automatically detected speech

Results obtained by the Filtering stage and the previous stages are summarized Figure 4. Among the 1950 audio events, 277 are distress calls, 45 are colloquial sentences and 1628 are non-speech. Considering the global performances, i.e. all the steps (discrimination, ASR and filtering modules), 61% of the distress calls uttered by the speakers lead to send an alert to the concerned service. This value is still relatively too low to consider a real use in real-world applications. One explanation could be the presence of noise in the speech. Thus 95 speech events were noisy, in most cases due to the superposition of the noise made by the person during the fall to the speech. Only 43.2% of speech events were classified as speech by the Discrimination module when they were noisy versus 86.3% for clean speech. It is also not surprising, all depend on the proportion of noise related to speech in term of energy or time. Moreover, for clean speech classified as speech, WER is 33.28%, versus 77.91% for noisy speech.

The presented ASR results were obtained using Sphinx3. We obtained better results using a more sophisticated ASR approach by modellling acoustic using SGMM and on manually annotated data from the same corpus. Acoustic models were based on subspace GMM (sGMM) with the KALDI toolkit (Vacher et al., 2015c).

In the experiments, the acoustic models were context-dependent classical three-state left-right HMMs. Acoustic features were based on Mel-frequency cepstral coefficients, 13 MFCC-features coefficients were first extracted and then expanded with delta and double delta features and energy (40 features). Acoustic models were composed of 11,000 context-dependent states and 150,000 Gaussians. The state tying is performed using a decision tree based on a tree-clustering of the phones.

In addition, off-line fMLLR linear transformation acoustic adaptation was performed. The acoustic models were trained on 500 hours of transcribed French speech composed of the ESTER 1&2 (broadcast news and conversational speech recorded on the radio) and REPERE (TV news and talk-shows) challenges as well as from 7 hours of transcribed French speech of the SWEET-HOME corpus (Vacher et al., 2015b) which consists of records of 60 speakers interacting in the smart home and from 28 minutes of the Voix-détresse corpus recorded by the LIG laboratory (Aman et al., 2013) which is made of records of speakers

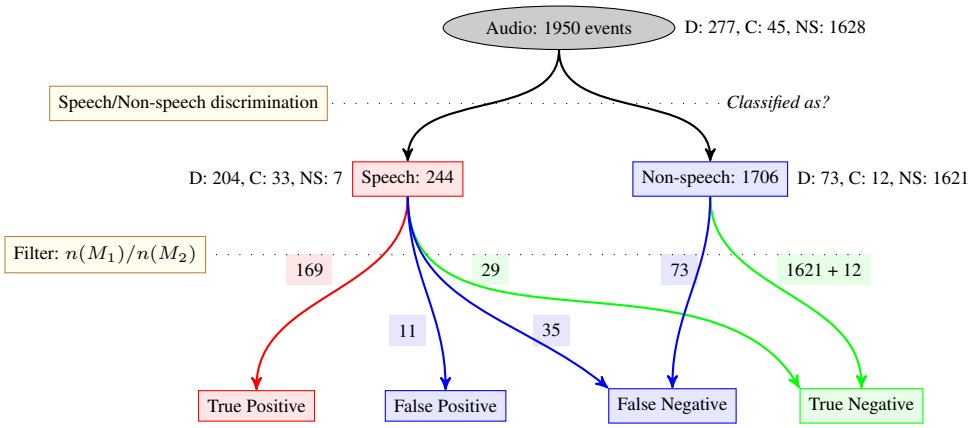

Figure 4: Global view of result experimentation in the DOMUS Living lab (Distress calls: D, Colloquial sentence: C, Non-Speech: NS).

eliciting a distress emotion. WER was 34% and 72% of the calls were detected using Kaldi and SGMM instead of 61% using Sphinx3 .

4. Conclusion

Audioprocessing in the Smart Homes, despite recent developments, have led to few experimentations in daily living conditions as far as audio analysis is concerned. This paper presents a modular audio analysis software able to analyse online the audio environment from multichannel, to extract the uttered sentences and then to process them thanks to an ASR stage. Moreover this system perfoms non-speech audio event identification; in this case, specific models must be trained. The software is designed to be modular and to process on-line the audio stream.

This software was used for call for help recognition in a smart home. The records were made in distant speech conditions, moreover the conditions were very harsh because the scenarios included falls of the person on the carpet. Audio events were correctly processed along the different stages. Results must be improved by using new models.

We plan to diffuse the software under the GNU GPL license.

Acknowledgements

This work was supported by the French funding agencies ANR and CNSA through CIRDO project (ANR-2010-TECS-012).

5. Bibliographical References

Aman, F., Vacher, M., Rossato, S., and Portet, F. (2013). Analysing the Performance of Automatic Speech Recognition for Ageing Voice: Does it Correlate with Dependency Level? In *4th Workshop on Speech and Language Processing for Assistive Technologies*, pages 9–15, Grenoble, France.

Audacity. (2016). `http://audacityteam.org/`. Online; accessed 29 January 2016.

Badii, A. and Boudy, J. (2009). CompanionAble - integrated cognitive assistive & domotic companion robotic systems for ability & security. In *1st Congres of the Société Française des Technologies pour l'Autonomie et de Gérontechnologie (SFTAG'09)*, pages 18–20, Troyes.

Bonastre, J.-F., Wils, F., and Meignier, S. (2005). Alize, a free toolkit for speaker recognition. In *Acoustics, Speech, and Signal Processing, 2005. Proceedings. (ICASSP '05). IEEE International Conference on*, volume 1, pages 737–740, March.

Bouakaz, S., Vacher, M., Bobillier-Chaumon, M.-E., Aman, F., Bekkadja, S., Portet, F., Guillou, E., Rossato, S., Desserée, E., Traineau, P., Vimon, J.-P., and Chevalier, T. (2014). CIRDO: Smart companion for helping elderly to live at home for longer. *IRBM - Ingénierie et Recherche Biomédicale*, 35(2):101–108, March.

CCRMA. (2016). Planet CCRMA at Home. `https://ccrma.stanford.edu/software/`. Online; accessed 29 January 2016.

Christensen, H., Casanueva, I., Cunningham, S., Green, P., and Hain, T. (2013). homeservice: Voice-enabled assistive technology in the home using cloud-based automatic speech recognition. In *4th Workshop on Speech and Language Processing for Assistive Technologies*, pages 29–34.

ChucK. (2016). ChucK : Strongly-timed, Concurrent, and On-the-fly Music Programming Language. `http://chuck.cs.princeton.edu/`. Online; accessed 29 January 2016.

Cristoforetti, L., Ravanelli, M., Omologo, M., Sosi, A., Abad, A., Hagmueller, M., and Maragos, P. (2014). The DIRHA simulated corpus. In *The 9th edition of the Language Resources and Evaluation Conference (LREC)*, pages 2629–2634, Reykjavik, Iceland.

Filho, G. and Moir, T. (2010). From science fiction to science fact: a smart-house interface using speech technology and a photorealistic avatar. *International Journal of Computer Applications in Technology*, 39(8):32–39.

Gemmeke, J. F., Ons, B., Tessema, N., Van Hamme, H., Van De Loo, J., De Pauw, G., Daelemans, W., Huyghe, J., Derboven, J., Vuegen, L., Van Den Broeck, B., Karsmakers, P., and Vanrumste, B. (2013). Self-taught assistive vocal interfaces: an overview of the aladin project.

In *Interspeech 2013*, pages 2039–2043.

Hamill, M., Young, V., Boger, J., and Mihailidis, A. (2009). Development of an automated speech recognition interface for personal emergency response systems. *Journal of NeuroEngineering and Rehabilitation*, 6(1):26.

Istrate, D., Castelli, E., Vacher, M., Besacier, L., and Serignat, J.-F. (2006). Information Extraction from Sound for Medical Telemonitoring. *IEEE Transactions on Information Technology in Biomedicine*, 10(2):264–274.

Istrate, D., Vacher, M., and Serignat, J.-F. (2008). Embedded implementation of distress situation identification through sound analysis. *The Journal on Information Technology in Healthcare*, 6:204–211.

Lamel, L., Gauvain, J., and EskEnazi, M. (1991). BREF, a large vocabulary spoken corpus for french. In *Proceedings of EUROSPEECH 91*, volume 2, pages 505–508, Geneva, Switzerland.

Levenshtein, V. I. (1966). Binary codes capable of correcting deletions, insertions and reversals. *Soviet Physics-Doklady*, 10:707–710.

Linguistic Data Consortium. (2011). French Gigaword Third Edition. https://catalog.ldc.upenn.edu/.

Povey, D., Ghoshal, A., Boulianne, G., Burget, L., Glembek, O., Goel, N., Hannemann, M., Motlicek, P., Qian, Y., Schwarz, P., Silovsky, J., Stemmer, G., and Vesely, K. (2011). The kaldi speech recognition toolkit. In *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding*. IEEE Signal Processing Society, December. IEEE Catalog No.: CFP11SRW-USB.

Sox. (2016). SoX - Sound eXchange. `http://sourceforge.net/projects/sox`. Online; accessed 29 January 2016.

Vacher, M., Istrate, D., and Serignat, J. (2004). Sound detection and classification through transient models using wavelet coefficient trees. In Suvisoft LTD, editor, *Proc. 12th European Signal Processing Conference*, pages 1171–1174, Vienna, Austria, sep.

Vacher, M., Fleury, A., Portet, F., Serignat, J.-F., and Noury, N. (2010). Complete Sound and Speech Recognition System for Health Smart Homes: Application to the Recognition of Activities of Daily Living. In Domenico Campolo, editor, *New Developments in Biomedical Engineering*, pages pp. 645 – 673. In-Tech, February. ISBN: 978-953-7619-57-2.

Vacher, M., Lecouteux, B., Istrate, D., Joubert, T., Portet, F., Sehili, M., and Chahuara, P. (2013). Experimental Evaluation of Speech Recognition Technologies for Voice-based Home Automation Control in a Smart Home. In *4th Workshop on Speech and Language Processing for Assistive Technologies*, pages 99–105, Grenoble, France, August.

Vacher, M., Lecouteux, B., Chahuara, P., Portet, F., Meillon, B., and Bonnefond, N. (2014). The Sweet-Home speech and multimodal corpus for home automation interaction. In *The 9th edition of the Language Resources and Evaluation Conference (LREC)*, pages 4499–4506, Reykjavik, Iceland, May.

Vacher, M., Aman, F., Rossato, S., and Portet, F. (2015a). Development of Automatic Speech Recognition Techniques for Elderly Home Support: Applications and Challenges. In J. Zou et al., editors, *Human Aspects of IT for the Aged Population. Design for the Everyday Life. First International Conference, ITAP 2015, Held as Part of the 17th International Conference on Human-Computer Interaction (HCII)*, volume Part II of *LNCS 9194*, pages 341–353, Los Angeles, CA, United States, August. Springer International Publishing Switzerland.

Vacher, M., Caffiau, S., Portet, F., Meillon, B., Roux, C., Elias, E., Lecouteux, B., and Chahuara, P. (2015b). Evaluation of a context-aware voice interface for Ambient Assisted Living: qualitative user study vs. quantitative system evaluation. *ACM Transactions on Accessible Computing*, 7(issue 2):5:1–5:36, May.

Vacher, M., Lecouteux, B., Aman, F., Rossato, S., and Portet, F. (2015c). Recognition of Distress Calls in Distant Speech Setting: a Preliminary Experiment in a Smart Home. In *6th Workshop on Speech and Language Processing for Assistive Technologies*, 6th Workshop on Speech and Language Processing for Assistive Technologies, pages 1–7, Dresden, Germany, September. SIG-SLPAT.

Vacher, M., Bouakaz, S., Chaumon, M.-E. B., Aman, F., Khan, R. A., and Bekkadja, S. (2016). The CIRDO corpus: comprehensive audio/video database of domestic falls of elderly people. In *LREC*, pages 1–8. (accepted paper).